Analyse Exploratoire Supervisée

Nous allons durant cette séance réaliser l'analyse exploratoire des données *Produit* afin d'identifier d'éventuels problèmes de qualité des données, de découvrir des indicateurs généraux sur les données et les propriétés de l'espace des données, et d'identifier les tâches de prétraitement des données nécessaires à la construction d'un l'ensemble de données aussi pertinent que possible.

1. Ensemble de données *Produit*

Caractéristiques de l'ensemble de données :

- Instances : 600 clients
- Nombre de variables : 12
- Valeurs manquantes : aucune
- Séparateur de colonnes : virgule
- Séparateur de décimales : point
- Variable de classe : Produit
- Variables prédictives : Age, Sexe, Habitat, Revenus, Marie, Enfants, Voiture, Compte_Epargne, Compte_Courant, Emprunt

Dictionnaire des données

Variable	Type	Description	Domaine de valeurs
ID	Entier	Numéro identifiant du client	[12101, 12400]
Age	Entier	Age en années	[18, 67]
Sexe	Catégoriel	Sexe	Homme, Femme
Habitat	Catégoriel	Type d'habitat	Centre_Ville, Petite_Ville, Rural, Banlieue
Revenus	Entier	Revenus annuels en dollars US	[60392, 505040]
Marie	Booléen	Statut marital	Oui, Non
Enfants	Entier	Nombres d'enfants	[0, 3]
Voiture	Booléen	Possède une voiture	Oui, Non
Compte_Epargne	Booléen	Possède un compte épargne	Oui, Non
Compte_Courant	Booléen	Possède un compte courant	Oui, Non
Emprunt	Booléen	Emprunt en cours	Oui, Non
Produit	Booléen	Client acquéreur du produit Variable de classe	Oui, Non

2. Chargement et exploration initiale des données

- Chargez les données du fichier `Data Produit.csv` dans un data frame `produit`.
- Vérifiez le chargement des données en affichant la liste des variables et leur mode à l'aide de la fonction `str()`.
- Affichez les noms des variables contenues dans le data frame `produit` à l'aide de la fonction `names()` par la commande :

```
> names(produit)
```

La fonction `summary()` appliquée à une (ou plusieurs) variable(s) renvoie des statistiques élémentaires sur celle(s)-ci en fonction de son type :

- Continue : valeur moyenne et bornes définissant les 4 quartiles (valeur minimale,, maximum du 1er quartile, médiane, maximum du 3ème quartile, valeur maximale).

• Discrète : nombre d'instances pour chaque valeur de la variable.

☛ Appliquez la fonction `summary()` au data frame `produit` par la commande :

```
> summary(produit)
```

☛ À l'aide de l'opérateur de sélection *`data_frame[selecteur_lignes, selecteur_colonnes]`*, affichez les instances du data frame `produit` appartenant à la classe *Produit = Oui* par la commande :

```
> produit[produit$Produit=="Oui", ]
```

☛ Appliquez la fonction `length(`*`expression`*`)` à la commande précédente. À quoi correspond la valeur retournée ?

☛ Affichez les identifiants (i.e. uniquement les valeurs de la variable *ID*) des instances appartenant à la classe *Produit = Oui*.

```
> produit[produit$Produit=="Oui", "ID"]
```

☛ Appliquez la fonction `length()` à la commande précédente.

À quoi correspond la valeur retournée ?

La fonction `table(`*`var`*`)` permet d'obtenir un affichage textuel des effectifs pour chaque valeur de la variable *`var`* reçue en paramètre. Cet affichage n'est donc utile que pour les variables discrètes (dont le nombre de valeurs est limité).

☛ Affichez les effectifs de chacune des valeurs de la variable `Produit` par la commande :

```
> table(produit$Produit)
```

☛ Affichez un graphique sectoriel des valeurs de la variable `Produit` par la commande :

```
> pie(table(produit$Produit), main = "Répartition des classes")
```

☛ Affichez les effectifs de chacune des valeurs de la variable `Habitat`.

☛ Affichez un graphique sectoriel des valeurs de la variable `Habitat`.

3. Histogrammes d'effectifs

Nous allons utiliser les fonctions de la librairie *ggplot2* afin d'afficher des diagrammes d'effectifs des attributs du data frame `produit`.

☛ Dans un navigateur Internet, allez sur le site http://docs.ggplot2.org/ qui décrit les nombreuses possibilités offertes par *ggplot2*.

Si besoin, vous y trouverez la description des commandes qu'il vous est demandé d'exécuter dans la suite afin de déterminer comment les paramétrer.

☛ Installez la librairie `ggplot2` et activez-la dans votre session R.

La commande `qplot(`*`var`*`, data=`*`data_frame`*`)` permet la création de diagrammes d'effectifs univariés (c-à-d monodimensionnels) afin d'observer la répartition des valeurs de la variable *`var`* dans le data frame *`data_frame`*.

☛ Affichez l'histogramme d'effectifs de la variable `Produit` du data frame `produit` par la commande :

```
> qplot(Produit, data=produit)
```

L'histogramme généré comporte une barre pour chaque valeur de *Produit* dont la hauteur correspond à l'effectif (nombre d'instances) de la valeur.

Cet histogramme nous permet de vérifier que les deux classes, *Produit=Oui* et *Produit=Non*, sont bien « suffisamment » fréquentes dans les données.

☛ Affichez l'histogramme d'effectifs de la variable `Sexe`, l'histogramme d'effectifs de la variable `Habitat`, et l'histogramme d'effectifs de la variable `Marie` afin d'observer la distribution de leurs valeurs.

Il est possible d'observer la répartition des classes pour chaque valeur de la variable (i.e. chaque barre) affichée dans l'histogramme.

Dans ce cas, une couleur est attribuée à chaque classe (e.g. *Produit=Oui* en vert et *Produit=Non* en rouge) et chaque barre de l'histogramme comporte alors une partie de chaque couleur dont la taille est proportionnelle à l'effectif de la classe pour cette valeur de la variable (barre).

Afin d'afficher la répartition des classes (*Produit=Oui* et *Produit=Non*) dans chacune des barres des histogrammes, il faut ajouter le paramètre `color=Produit` à la commande `qplot()`.

☛ Affichez l'histogramme d'effectifs de la variable `Habitat` en affichant en couleur la répartition des classes pour chaque barre.

Dans chaque barre, cet affichage colore les contours des deux zones qui représentent les effectifs des classes (une couleur pour *Produit=Oui* et une autre couleur pour *Produit=Non*).

☛ Afin d'obtenir un affichage plus lisible, remplacez le `color=Produit` par le paramètre `fill=Produit` dans la commande `qplot()` précédente.

Cet affichage colore l'intérieur des deux zones de chaque barre qui représentent les effectifs des classes, rendant le graphique plus lisible.

La variable `Enfants`, bien que numérique, donnera lieu à un affichage similaire du fait de son faible nombre de valeurs (*0*, *1*, *2* ou *3*) qui fait que la fonction l'affiche comme une variable discrète.

☛ Affichez l'histogramme d'effectifs de la variable `Enfants` en affichant en couleur la répartition des classes pour chaque barre.

☞ Voyez-vous des différences notables dans les proportions des classes entre les valeurs de `Enfants` ?

☞ Identifiez pour chaque valeur de `Enfants` la classe la plus fréquente.

Dans le cas des autres variables numériques (e.g. `Age` ∈ [18,67]), la variable est d'abord discrétisée, c'est-à-dire que son domaine de valeurs est divisé en intervalles (e.g. intervalles `Age` ∈ { [18:30], [31:54], [55:67] }).

Une barre est alors affichée dans l'histogramme pour chaque intervalle et sa hauteur est proportionnelle au nombre d'exemples dont la valeur pour la variable est dans cet intervalle.

☛ Affichez l'histogramme d'effectifs de la variable `Age` et l'histogramme d'effectifs de la variable `Revenus` afin d'observer la répartition des valeurs des variables dans le domaine de valeurs.

☞ Voyez-vous des différences notables dans les proportions des classes entre les valeurs des variables ?

☛ Affichez l'histogramme d'effectifs de la variable `Revenus` en affichant en couleur la répartition des classes (paramètre `fill`) et en fixant la largeur des barres de l'histogramme à 40 000 à l'aide du paramètre `binwidth`.

Chaque barre correspond maintenant à un intervalle de 40 000$ de `Revenus` (e.g. de 60 392 $ à 100 392 $ pour la première barre).

☛ Afin d'observer les différences dans la répartition des classes pour des valeurs « très faibles », « faibles », « moyennes », « élevées » et « très élevées » de `Revenus`, affichez l'histogramme d'effectifs de `Revenus` avec :

☞ En couleur la répartition des classes.

☞ En fixant le nombre de barres de l'histogramme à 5 à l'aide du paramètre `bins`.

☛ Affichez l'histogramme d'effectifs de la variable `Age` en affichant en couleur la répartition des classes et en fixant la largeur des barres de l'histogramme à 3 à l'aide du paramètre `binwidth`.

☛ Affichez l'histogramme d'effectifs de `Age` en affichant en couleur la répartition des classes et en fixant la nombre de barres de l'histogramme à 5 à l'aide du paramètre `bins`.

4. Nuages de points

La fonction `qplot()` de la librairie *ggplot2* permet notamment d'afficher des nuages de points (diagrammes multivariés, c-à-d multidimensionnels).

Chaque exemple est alors représenté par un point positionné dans le graphique selon ses valeurs pour les variables représentées en abscisse et en ordonnée du graphique.

La commande `qplot(`*`var1`*`, `*`var2`*`, data=`*`nom_data_frame`*`)` permet la création d'un nuage de points avec pour axe des abscisses la variable *`var1`* et pour axe des ordonnées la variable *`var2`* à partir du data frame *`nom_data_frame`*.

☛ Affichez un nuage de points avec pour axe des abscisses la variable `Age` et pour axe des ordonnées la variable `Revenus`.

Le paramètre `color=`*`var3`* ajouté à la commande `qplot()` précédente permet de colorer les points en fonction des valeurs de la variable *`var3`*.

☛ Modifiez la commande précédente afin que la couleur des points représente la classe de l'instance : *Produit=Oui* ou *Produit=Non*.

Le nuage de points affiché permet de constater que si l'âge et les revenus du client sont faibles alors le

client a plus de chances de ne pas acheter le produit : la majorité des points dans la partie inférieure gauche du graphique sont de la classe *Produit=Non* alors que la majorité des points dans la partie supérieure droite du graphique sont de la classe *Produit=Oui*.

Dans le cas de l'affichage de variable discrètes, il est nécessaire d'ajouter un déplacement aléatoire des points afin de tous les distinguer, car certains points ont exactement la même position dans le graphique.

☛ Affichez un nuage de points avec pour axe des abscisses la variable `Revenus`, pour axe des ordonnées la variable `Enfants` et en couleur la classe des instances.

Tous les points correspondant à la même valeur de la variable discrète `Enfants` étant situés sur la même ligne, nous allons utiliser l'instruction `+ geom_jitter()` afin d'ajouter un léger déplacement aléatoire des points pour les distinguer tous.

☛ Ajoutez un déplacement aléatoire vertical des points en ajoutant après la commande précédente le paramètre `+ geom_jitter(height = 0.1)` par la commande :

```
> qplot(…) + geom_jitter(height = 0.1)
```

☛ Ajustez la valeur du paramètre `height` définissant l'amplitude maximale du déplacement afin d'obtenir l'affichage le plus lisible (valeur entre 0.1 et 0.4).

Pour chaque ligne, nous pouvons voir apparaître des zones constituées quasi-exclusivement de points de la même couleur.

Cela signifie que pour le nombre d'enfants correspondant à cette ligne, à partir d'une certaine valeur de `Revenus`, la majorité des instances appartiennent à la même classe.

☛ Pour les valeurs `1`, `2` et `3` de `Enfants`, estimez (en déterminant visuellement la valeur approximative) la valeur de `Revenus` à partir de laquelle toutes les instances ou presque sont de la classe *Produit=Oui*.

Si les deux variables affichées sont discrètes (non continues), il est nécessaire de définir à la fois un déplacement vertical et un déplacement horizontal des points.

☛ Affichez un nuage de points avec la variable *Mari*e pour axe des abscisses, la variable `Enfants` pour axe des ordonnées et en couleur la classe des instances.

Seuls huit points apparaissent, chacun correspondant à une combinaison de valeurs pour `Marie` et `Enfants` (e.g. *Marie=Oui* et *Enfants=0, Marie=Non* et *Enfants=0*).

☛ Ajoutez des déplacements aléatoires vertical et horizontal des points en ajoutant après la commande précédente le paramètre `+ geom_jitter(height = 0.1, width = 0.1)` par la commande :

```
> qplot(…) + geom_jitter(height = 0.1, width = 0.1)
```

☛ Ajustez la valeur des paramètres `height` et `width` définissant l'amplitude maximale des déplacements vertical et horizontal afin d'obtenir l'affichage le plus lisible.

☛ Quelle information déduisez-vous de ce nuage de points pour les personnes sans enfants ?

5. Boîtes à moustaches

La fonction `boxplot()` de R permet la création de boîtes à moustaches (boxplots) afin d'observer la distribution des valeurs d'une variable.

Elle permet de réaliser un affichage graphique des informations renvoyées par la fonction `summary()`.

Elle permet également de comparer les distributions des valeurs d'une variable pour chaque valeur d'une autre variable (e.g. la variable de classe).

La commande `boxplot(`*`var`*`, data=`*`data_frame`*`)` permet de créer une boîte à moustaches pour observer la distribution des valeurs de la variable *`var`* dans le data frame *`data_frame`*.

☛ Créez une boîte à moustaches afin d'afficher la distribution des valeurs de la variable `Age` dans le data frame `produit`.

La boîte centrale contient les 50 % de valeurs centrales, c-à-d entre la valeur minimale du 2ème quartile et la valeur maximale du 3ème quartile.

Les deux moustaches contiennent chacune 25 % des valeurs les plus faibles et les plus élevées respectivement.

Nous pouvons observer une distribution homogène des valeurs, c-à-d des tailles proches pour les quatre différentes zones affichées, correspondant chacune à un quartile.

Consultez la page http://www.statmethods.net/graphs/boxplot.html pour plus de détails sur la fonction `boxplot()` et ses paramètres.

- Modifiez la commande précédente afin d'afficher pour titre du diagramme « Distribution de Age » et pour nom de l'axe des ordonnées « Valeur de Age ».
- Utilisez la fonction `summary(variable)` afin d'identifier les valeurs de `Age` correspondant à la médiane et aux valeurs minimale et maximale de la boîte (c-à-d la valeur minimale du 2ème quartile et la valeur maximale du 3ème quartile).
- Créez une boîte à moustaches afin d'afficher la distribution des valeurs de la variable `Revenus` dans le data frame `produit`.

  ☞La distribution des valeurs de `Revenus` vous paraît-elle homogène ?
- Utilisez la fonction `summary()` afin d'identifier la valeur médiane et les valeurs minimale et maximale de l'intervalle contenant les 50 % de valeurs centrales de `Revenus`.

La commande `boxplot(var1~var2, data=data_frame)` permet de créer une boîte à moustaches pour comparer la distribution des valeurs de la variable *var1* pour chaque valeur (ou intervalle de valeurs si elle est numérique continue) de la variable *var2*.

- Créez une boîte à moustaches afin de comparer la distribution des valeurs de la variable `Age` pour chaque valeur de `Produit` dans le data frame `produit`.

  Affichez pour titre du diagramme « Age selon Produit », pour nom de l'axe des ordonnées « Age » et pour nom de l'axe des abscisses « Produit ».
- Exécutez la commande suivante afin d'afficher les statistiques élémentaires de `Age` pour chacune des deux classes, c-à-d appliquer la fonction `summary()` pour chaque combinaison de valeurs des deux variables `Age` et `Produit` :

```
> tapply(produit$Age, produit$Produit, summary)
```

  ☞Identifiez la valeur médiane, et les valeurs minimale et maximale de l'intervalle contenant les 50 % de valeurs centrales pour chacune des deux classes.
- Créez une boîte à moustaches afin de comparer la distribution des valeurs de la variable `Revenus` pour chaque valeur de `Produit` dans le data frame `produit`.

  Utilisez le paramètre `col=c("tomato","darkturquoise")` afin d'afficher en turquoise la boîte pour *Produit=Oui* et en rouge tomate la boîte pour *Produit=Non*.

Nous pouvons observer des distributions différentes des valeurs de `Revenus` pour les deux classes car les positions verticales des 2 boîtes, et de leurs moustaches, sont décalées.

- Affichez les statistiques élémentaires de `Revenus` pour chacune des deux classes (application de la fonction `summary` pour chaque combinaison de valeurs des deux variables `Revenus` et `Produit`).

  ☞Identifiez la valeur médiane et la moyenne de `Revenus` pour les exemples *Produit=Oui* et les exemples *Produit=Non*.

6. Tables de Contingence

La fonction `table(var1, var2)` de R Base permet d'obtenir un affichage textuel sous forme de table des effectifs pour chaque combinaison de valeurs des variables discrètes reçues en paramètre (table de contingence).

- Affichez les effectifs de chaque valeur de la variable catégorielle `Habitat` pour chaque valeur de la variable `Produit` par la commande :

```
> table(produit$Habitat, produit$Produit)
```

- Affichez les effectifs de chaque valeur de la variable `Enfants` pour chaque valeur de la variable `Produit`.
- Affichez les effectifs de chaque valeur de la variable binaire `Sexe` pour chaque valeur de la variable `Produit`.
- Afin de faciliter la lecture des résultats décimaux suivants, définissez le nombre de décimales qui seront affichées par la suite avec la commande :

```
> options(digits=2)
```

La fonction `prop.table(table(var1, var2))` de R Base permet d'obtenir un affichage des effectifs pour chaque combinaison de valeurs des variables discrètes *var1* et *var2* reçues en paramètre (table de contingence) sous forme de proportions.

- Affichez la table de contingence en proportions des effectifs pour chaque combinaison de valeurs de

`Habitat` et `Produit` par la commande :

```
> prop.table(table(produit$Habitat, produit$Produit, dnn=c("Habitat",
  "Produit")))
```

- Affichez cette même table de contingence en pourcentages d'effectifs par la commande :

```
> prop.table(table(produit$ Habitat, produit$Produit, dnn=c("Habitat",
  "Produit")))*100
```

- Affichez la table de contingence en proportions des effectifs pour chaque combinaison de valeurs de `Enfants` et `Produit`.
- Affichez la table de contingence en pourcentages d'effectifs pour chaque combinaison de valeurs de `Enfants` et `Produit`.
- Affichez la table de contingence en proportions des effectifs pour chaque combinaison de valeurs de `Marie` et `Produit`.
- Affichez la table de contingence en pourcentages d'effectifs pour chaque combinaison de valeurs de `Marie` et `Produit`.

La fonction `mosaic()` de la librairie *regclass* permet d'afficher un graphique en mosaïque des proportions de valeurs de deux variables discrètes.

- Affichez un graphique en mosaïque des effectifs pour chaque combinaison de valeurs des variables `Habitat` et `Produit` du data frame `produit` par la commande :

```
> mosaic(Habitat~Produit, data=produit, color=TRUE)
```

  - ☞Observez-vous des combinaisons de valeurs de `Habitat` et `Produit` notables, c-à-d qui sont sur-représentées ou sous-représentées d'après la taille des rectangles dans le graphique obtenu ?
- Affichez un graphique en mosaïque des effectifs pour chaque combinaison de valeurs des variables `Enfants` et `Produit`.
  - ☞Quelles combinaisons de valeurs de `Enfants` et `Produit` sont notables, c-à-d sur-représentées ou sous-représentées, selon la taille des rectangles dans le graphique obtenu ?
- Affichez un graphique en mosaïque des effectifs pour chaque combinaison de valeurs des variables `Marie` et `Produit`.